Haubro, M., Orfanidis, C., Oikonomou, G., & Fafoutis, X. (2020). TSCH-over-LoRA: Long Range and Reliable IPv6 Multi-hop Networks for the Internet of Things. *Internet Technology Letters*, *3*(4), [e165]. https://doi.org/10.1002/itl2.165

Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

TSCH-over-LoRA: Long Range and Reliable IPv6 Multi-hop Networks for the Internet of Things

Martin Haubro[1] | Charalampos Orfanidis[2] | George Oikonomou[3] | Xenofon Fafoutis*[1]

[1]Technical University of Denmark, Denmark
[2]KTH Royal Institute of Technology, Sweden
[3]University of Bristol, UK

Correspondence
*Corresponding author, Email: xefa@dtu.dk

Summary

TSCH-over-LoRa is a long range and reliable IPv6 multi-hop solution that aims at combining the reliability of TSCH (Time-Slotted, Channel Hopping) together with the long range capabilities of LoRa. TSCH-over-LoRa brings mesh IPv6 networking to LoRa devices, enabling the use of standard protocols (such as RPL, UDP, and CoAP) and long range operation to TSCH/6TiSCH industrial wireless IoT networks. We design, implement, and integrate TSCH-over-LoRa into the TSCH/6TiSCH networking stack of the Contiki-NG operating system and experimentally demonstrate its compatibility with higher-level protocols and its resilience to interference.

KEYWORDS:
TSCH, LoRa, Industrial IoT, Mesh Networks, 6LoWPAN, Internet of Things

1 | INTRODUCTION

LoRa[1] is a low-power, long range radio technology that emerged in 2015 and is considered an enabler of Internet-of-Things (IoT) networks over long distances, enabling all sorts of sensing applications: from city-wide power-grids and smart meters to urban air-pollution or reindeer tracking. The current Medium Access Control (MAC) standard for using LoRa is LoRaWAN[2], which organises the network in a star topology, and assumes that every radio is always within range of a LoRaWAN gateway. While this makes a case for extreme low-power usage and is intuitive, it also provides with some limitations, such as not having any opportunity of doing multi-hop mesh networking. While LoRa has shown ranges of several kilo-meters with clear line of sight in best-case scenarios, it can still be argued that not supporting mesh networking puts a limitation to the technology, as multi-hop networking can greatly enhance the possible ranges that a sensing network can cover. This can, in particular, make sense in scenarios where it is not easy to deploy powered LoRaWAN-gateways, like e.g. arctic scenarios. Moreover, the LoRaWAN stack is notoriously unreliable and vulnerable to collisions[3], as well as vulnerable to cross-technology interference[4].

For wireless reliability, one has to look at industrial wireless standards, such as TSCH (Time-Slotted, Channel Hopping). TSCH is a MAC protocol that traces its roots to traditional industrial wireless standards, such as WirelessHart[5] and ISA100[6], and has been recently introduced in the IEEE 802.15.4 standard[7]. TSCH has been shown to provide very high reliability in real-world deployments[8], as well as deterministic delays[9]. It achieves this by keeping the nodes time-synchronised and orchestrating transmissions using a schedule. A TSCH schedule can be free from internal collisions, as long as the scheduler allocates only a single transmitter to each timeslot. Moreover, TSCH is resilient to external interference and multi-path fading by employing channel hopping. Building on foundations of TSCH, 6TiSCH is a standardisation effort by IETF that provides a full IPv6 stack for industrial IoT applications[10]. The stack has been ported in several operating systems for IoT devices, such as Contiki-NG[11].

In this paper, we present TSCH-over-LoRa: A long range and reliable IPv6 multi-hop solution that uses TSCH at the MAC layer and LoRa at the physical layer, combining the reliability of the former with the long-range operation of the latter. The novelty of the scheme lies in the unique combination of TSCH and LoRa, resulting to a protocol stack that is different from both

traditional LoRa networks and traditional TSCH networks. TSCH-over-LoRa brings long range operation to TSCH/6TiSCH industrial wireless IoT networks and mesh networking to LoRa networks. Furthermore, TSCH-over-LoRa brings IPv6 to LoRa devices, enabling the use of standard protocols, such as IETF RPL and IETF CoAP. This addresses niche applications, such as ultra-low bandwidth applications that require high reliability and long-range operation and ultra-long-range monitoring applications that require more than one LoRa links. We provide a full implementation of TSCH-over-LoRa for Contiki-NG†.

This letter is organised as follows. Section 2 summarises the related work. Section 3 provides design and implementation details of TSCH-over-LoRa. Section 4 evaluates it experimentally. Finally, Section 5 provides concluding remarks.

2 | RELATED WORK

This work adds a new wireless technology, namely LoRa, to an IPv6/6LoWPAN stack. This is in similar spirit to BLEACH[12], which implements IPv6/6LoWPAN over Bluetooth Low Energy (BLE). Likewise, UWB-TSCH[13] focuses on Ultra Wide Band (UWB) communications.

Previous work has studied the limitations of LoRaWAN[3], including the number of collisions when many LoRa devices generate data frequently, assuming the ALOHA-access in LoRaWAN where devices may transmit anytime. This also shows that LoRa scales badly with a high amount of end-devices due to collisions. Further work[14] studied the orthogonality of spreading factors, and suggested LoRaBlink, which is a protocol that aims to provide reliable and efficient multi-hop networking, and bi-directional communication. Like TSCH, LoRaBlink uses timeslots and beacons for synchronisation, however, unlike TSCH, it is not using channel hopping, and offers no option for scheduling, other than transmitting in the next slot. Time synchronisation is also never done on receiving acknowledgements, rather, the whole network synchronises once every synchronisation period. Yet, they also achieve multi-hop connectivity and a reliability of 80%.

KRATOS[15] provides a hardware-software platform for LoRa research. KRATOS provides embedding drivers for integrating a different LoRa device, namely the SX1276, into the Contiki operating system. However, currently, KRATOS is limited to only supporting MSP430 and Tmote Sky, and their LoRa driver is implemented in a platform-dependent manner. This technically brings LoRa to IPv6 networks, but only for a single chip family, and not focused on using TSCH, rather, the implementation is focused on building a wake-on-interrupt LoRa radio, and is therefore unable to poll or deliver timestamps.

An extension to LoRaWAN allowing for multi-hop LoRaWAN has also been proposed[16]. However, the authors still rely on the LoRaWAN gateways, and their relay nodes are assumed to be mains-powered, unlike our approach. In addition, channel hopping is not used. Their time synchronisation mechanism is quite similar to that of TSCH, using beacons to synchronise on a regular basis.

Our system integrates LoRa into the TSCH/6TiSCH stack of Contiki-NG[11] and uses the Orchestra scheduler[17]. Moreover, we employ Contiki-NG's microsecond-level synchronisation method[18]. LoRa and standard-compliant TSCH have, in general, not been previously combined; however, previous work has built a TSCH-like setup with LoRa[19]. Yet, rather than building it upon a standard-compliant implementation, the authors craft it ground up, designing timeslots to contain different spreading factors, allowing communication across these in a single timeslot. They measure relatively low packet error rates, however, no energy measurements are made and time on air is not computed or mentioned.

3 | DESIGN AND IMPLEMENTATION

Our system is based on the SX1272 LoRa radio, though it can be trivially extended to other LoRa radios. The LoRa radio is integrated into the Contiki-NG system by building a radio driver that fits into the Contiki-NG networking stack, exposing the interfaces needed for the above layer, *i.e.* the MAC layer, to work, as well as interfacing to the Hardware Abstraction Layer (HAL) library, ensuring portability across platforms as well as support for above layers. The Contiki-NG driver stack is used for SPI connections as well as GPIO access to the reset-pin of the radio. The driver has been organised into three separate layers, as seen in Figure 1; one layer interfaces to the SPI and GPIO layers; one layer contains the main driver software, and one layer implements the external interface to *e.g.* the MAC layer. This approach aims at making the long-term maintenance of the driver easier, if *e.g.* Contiki-NG changes any of the above or below layers. It also makes it possible to port the SX1272 driver

† Available at https://github.com/dtu-ese/contiki-ng-lora.

Application/Transport Layer	CoAP/UDP
Network Layer	RPL, IPv6, 6LoWPAN
MAC Layer	TSCH, Orchestra Scheduler
Radio Driver Layer	Contiki Networking Interface
	Poll-Mode RX Driver for SX1272
	FIFO, SPI I/O
Hardware Abstraction Layer	SPI HAL
Device Radio	SX1272 Radio

FIGURE 1 LoRa integrated within the Contiki-NG IPv6 networking stack.

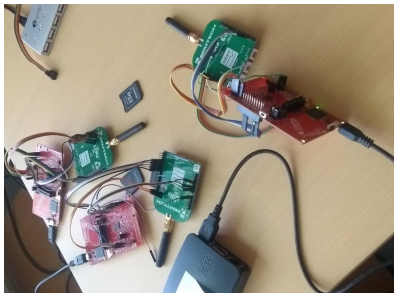

FIGURE 2 Testbed: SX1272 LoRa on CC1350.

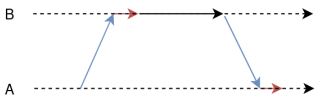

FIGURE 3 Rx IRQ delay modelling: the red arrows indicate the IRQ delay, blue is transmissions, and black is the wait time.

to other operating systems, as only those layers need adaptation. Figure 2 shows our experimental testbed: the SX1272 LoRa radio interfaced to a CC1350 Launchpad running Contiki-NG.

The SX1272 LoRa radio does not provide any active way to leave timestamps on received packets. This is an issue, as the time synchronisation that TSCH is built upon, requires knowledge of exactly when a packet arrived. Apart from this, TSCH requires a poll-mode driver – in part to be able to turn off the radio as soon as the packet is in the air, in part because TSCH itself runs in an interrupt context. This also means that the assumption can be made that the radio driver is constantly being polled for new information while actively receiving messages, and therefore the internal IRQ (Interrupt Request) of the radio is polled, and when an interrupt fires on the radio, the time is noted, and timestamp reverse-calculated. Polling whether there is a packet in the air is done by polling a status-register of the modem. The IRQ-interrupts are slightly delayed, depending on the setup of the radio, and so is the status register updates. These delays need to be known for the optimisation of the guard time. These estimations have been done by simulating clock synchronisation between two nodes, in having node A transmitting a message to node B, and whenever node B receives it, it starts transmitting a message to node A at exactly a given time after the timestamp node B noted it. Then A can look at the timestamp of the received packet, which should have two reception delays, and use this to estimate the IRQ delay, as seen in Figure 3. The delay of the modem's status register can be estimated by looking at when the radio notifies that there is a packet in the air, and compare it to the timestamps that can now be accurately assessed due to the discussed clock synchronisation simulation between the two nodes. These delays have turned out to be, for SF7 (125 kHz), roughly 800 μs on the IRQ register and at most 6000 μs on packet-in-air detection, and for SF10 (125 kHz), roughly 6250 μs on the IRQ detection, and 28000 μs on packet-in-air detection. These are, in practice, added to the guard time, as the radio in an Rx slot should not be turned off before it is confirmed there is no packet in the air, and if a transmission is ongoing the radio should not be turned off before it can be confirmed that the transmission has ended.

A TSCH timeslot is composed of the guard time, the packet transmission, and the reception of the acknowledgement. The guard time defines the resilience in synchronisation errors and, in turn, the required frequency of synchronisation events. In principle, it must be sufficiently large to provide resistance against clock drift, but small enough to conserve energy.

The total timeslot length is also dependent on the length of the longest possible transmission as well as longest possible acknowledgement, which is, in turn, dependent on the maximum payload size and the configuration parameters of LoRa, namely the spreading factor, the bandwidth, and the coding rate. Here, we provide the design of timeslots for SF7 and SF10, yet the same methodology can be used for extending it to all remaining spreading factors. Table 1 summarises the LoRa configuration parameters that our TSCH timeslots assume. Furthermore, in line with default settings of Contiki-NG TSCH, the channel hopping sequence is hard-coded.

Synchronisation in TSCH is done through sending out Enhanced Beacons (EB). Every member of the TSCH network sends out EBs on a regular interval. Nodes, however, also synchronise whenever they receive an acknowledgement from their time parent. The standard choice of Contiki-NG is that EBs are sent out every 12.5 seconds, however, due to LoRa duty cycling

Payload Size	100 bytes
Spread Factor (SF)	SF7/SF10
Explicit header	yes
Low data rate optimiser	no
Coding Rate (CR)	4/5
Preamble Symbols	6
Bandwidth (BW)	125 Hz

TABLE 1 LoRa Configuration Parameters.

Spreading Factor	SF7	SF10
TX Offset	7	37
RX Offset	5	5
RX Wait	8	64
Max TX Length	209	1438
Rx Ack Delay	5	15
Tx Ack Delay	3	25
Max Ack Length	10	365
Ack Wait	3	20
Total Timeslot Length	278	4500

TABLE 2 TSCH timeslot timings. All values are in ms.

constraints and the much longer timeslots, we opt for sending out EBs approximately every 30 seconds, and, as a minimum send one every 40 seconds. Having a synchronisation event once every 30 seconds, and considering 80 ppm maximum drift would require a guard time of at least 2400 μs. We opt for a slightly tighter value and set the guard time at 2000 μs. The final configuration values for a SF7 and SF10 TSCH timeslot are provided in Table 2. A TSCH-over-LoRa timeslot is 278 ms for transmitting a 127-byte packet that requires 209 ms on air (SF7). Assuming for simplicity that a single channel is used, this allows for scheduling approximately 12949 packets per hour without collisions. In LoRaWAN networks of more than 50 nodes, in comparison, less packets per hour are received due to collisions [20].

4 | EVALUATION

The evaluation of the system is done experimentally. The experiments aim to demonstrate the resiliency of TSCH-over-LoRa to interference, as well as its compatibility with low-power standards. Moreover, we assess the reliability and the radio duty cycle. All experiments are done using SX1272 connected to the CC1350 Launchpad over SPI (see Figure 2). The evaluation includes a 'base experiment', using simple UDP sockets to send messages from 2 nodes in a multi-hop network to a root node using SF7, along with different modifications of this, including using SF10, CoAP, and Ping. Moreover, we test the resilience of the network to interference using a jammer node. Channel hopping is only used in the jamming experiment, as it greatly increases potential scanning speed, setting up the TSCH network. In all cases, the radio duty cycle is measured from the time all nodes are reachable by the root node. In other words, we exclude the Rx-heavy scanning process of TSCH, as well as the initial period of the network, where some nodes are not part of the network yet.

For all experiments, the security layer of IEEE 802.15.4 is activated, and Orchestra [17] has been used as a scheduler to minimise collisions in slots when transmitting. TSCH time synchronisation beacons are being broadcasted at a 200 s interval, two orders of magnitude less frequently than the default configuration. RPL-lite has been used for routing, where it is been ensured that routing messages are sent much less frequently than the default configuration, due to the long transmission times and timeslots, making a single transmission more costly in terms of energy. All experimental results were recorded for an interval of at least 60 minutes, with at least 15 minutes of warm-up to setup the network, as there is a one-time cost when building the network.

4.1 | Channel Hopping LoRa Experiment

This experiment demonstrates the resilience of TSCH-over-LoRa to interference via channel hopping. The experiment is composed of two nodes, the leaf and the root, whereby the leaf node sends UDP messages to the root. The messages are 50-byte packets, sent at an interval of once per minute. Meanwhile, a third node is constantly sending out signals on a channel, effectively jamming it and making it unusable. Our goal is then to see whether the full system avoids the interference by re-transmitting the messages on different, non-jammed channels. The jammer is created by having a node continuously transmit messages with an interval of 1 tick, the equivalent of 1/128th of a second, between messages. The message is a short 1-byte message, which is done to ensure that as many preambles as possible are sent, that the other two nodes might pick up and listen to. The transmission of a 1-byte packet takes approximately 23.8 ms, and waiting is roughly 8ms. As such, the channel is occupied 75% of the

time. This experiment is first carried out first with channel hopping on 4 different channels, *i.e.* the jammed channel is used 25% of the time. For benchmarking, we compare this with a single-channel setting, whereby the two nodes always use the jammed channel. This emulates the traditional single-channel LoRa in a situation of high interference.

In a 70-minute experiment, TSCH-over-LoRA did not lose any packets, that is 100% Packet Delivery Rate (PDR). In the experiment with no channel hopping, it took 30 minutes for the leaf node to receive an EB and join the TSCH network. From there, after an additional hour, it failed to join the routing network, as it was not able to pick up the packets necessary for this, which means that no packets had been sent by the leaf, as there was seemingly no connection to the root node.

4.2 | Multi-Hop LoRa Experiments

In this series of experiments, we demonstrate multi-hop LoRa in various settings, and we measure the reliability (in terms of PDR) and the radio duty cycle (a proxy for energy consumption). These experiments are composed of three nodes in a multi-hop setting: a root node, a hop node, and a leaf node. In this setting both the leaf and hop nodes generate traffic periodically for the root node. In addition, the hop node forwards the packets of the leaf node in addition to the locally generated ones. The radio duty cycle is measured on the hop and leaf nodes, as the root node is assumed to be always on. The radio duty cycle is measured using the Energest tool of Contiki-NG. In this setting, we conduct the following experiments: (i) UDP packets (88 bytes IP payload) are generated every 5 minutes; (ii) CoAP packets (73 bytes IP payload), containing the node ID, are polled every 4 or 8 minutes from the leaf and hop nodes (we consider both confirmable and non-confirmable, nc, requests); and (iii) the root node pings the leaf and hop nodes every 4 or 8 minutes (68 bytes IP payload). All above experiments use SF7. As a final experiment we also generate UDP packets (88 bytes IP payload) every 10 minutes using SF10. The results are summarised in Table 3.

Scenario	SF	Period	Duration	Hop, Tx	Hop, Rx	Leaf, Tx	Leaf, Rx	Hop, PDR	Leaf, PDR
UDP	SF7	5m	1h35m	0.33%	1.42%	0.19%	1.33%	100%	100%
CoAP (nc)	SF7	8m	5h51m	0.23%	1.41%	0.13%	1.32%	99.77%	99.77%
CoAP	SF7	8m	2h54m	0.9%	1.74%	0.44%	1.46%	100%	100%
CoAP	SF7	4m	2h55m	1.79%	2.16%	0.86%	1.67%	100%	100%
Ping	SF7	8m	2h55m	0.93%	1.76%	0.45%	1.48%	100%	100%
Ping	SF7	4m	2h50m	0.42%	1.5%	0.15%	1.37%	100%	100%
UDP	SF10	10m	1h40m	2.35%	2.79%	1.36%	2.17%	100%	88.8%

TABLE 3 Results of Multi-Hop LoRa Experiments.

The results demonstrate that TSCH-over-LoRa is out-of-the-box compatible with the Orchestra TSCH scheduler and several higher-level protocols of Contiki-NG, including RPL, UDP, CoAP and Ping. The experiments also show the multi-hop LoRa with TSCH-over-LoRa can sustain a respectable packet throughput (a packet every less than 10 minutes) whilst respecting the 1% transmit radio duty cycle policy even at the hop node. However, it is also clear that high level protocols, such as CoAP with confirmable messages, introduce significant overhead that limits the throughput. The results also indicate that TSCH-over-LoRa is reliable; however this is demonstrated with a small sample set. Due to the duration of the experiments, it is, indeed, impractical to collect thousands of packets that would be required for measuring the PDR with high accuracy. Using SF7 in multi-hour experiments only a single packet was lost with the majority of the cases yielding 100% PDR. Using SF10, the reliability appears to suffer, however, more experiments are needed to further validate this due to the small sample size (10 packets are transmitted in under 2 hours) and confirm the robustness of the timings of the SF10 timeslot.

5 | CONCLUSION

TSCH-over-LoRa leverages the reliable operation of TSCH and the long-range capabilities of LoRa to provide a reliable and long range IPv6 multi-hop solution for IoT networks. In this work, we design and integrate TSCH-over-LoRa in the TSCH/6TiSCH networking stack of Contiki-NG and we experimentally demonstrate its resiliece to interference, its support for multi-hop operation and its off-the-shelf compatibility with IPv6-based protocol standards, such as RPL, UDP and CoAP.

References

1. Semtech . LoRa modulation basics. https://www.semtech.com/uploads/documents/an1200.22.pdf; 2019.
2. LoRa Alliance . What is the LoRaWAN® Specification?. https://lora-alliance.org/about-lorawan; 2019. [Online; accessed June 19, 2019].
3. Adelantado F, Vilajosana X, Tuset-Peiro P, Martinez B, Melià-Seguí J, Watteyne T. Understanding the limits of LoRaWAN. *IEEE Communications Magazine* 2017; 55. doi: 10.1109/MCOM.2017.1600613
4. Orfanidis C, Feeney LM, Jacobsson M, Gunningberg P. Investigating interference between LoRa and IEEE 802.15. 4g networks. In: IEEE. ; 2017: 1–8.
5. Chen D, Nixon M, Mok A. *WirelessHART: Real-Time Mesh Network for Industrial Automation*. Springer . 2010.
6. ISA100: Wireless Systems for Industrial Automation-Developing a Reliable, Universal Family of Wireless Standards. 2007.
7. IEEE Standard for Local and metropolitan area networks—Part 15.4. IEEE Std 802.15.4-2015; 2015.
8. Elsts A, Fafoutis X, Woznowski P, et al. Enabling Healthcare in Smart Homes: The SPHERE IoT Network Infrastructure. *IEEE Communications Magazine* 2018; 56(12): 164-170. doi: 10.1109/MCOM.2017.1700791
9. Koutsiamanis RA, Papadopoulos GZ, Fafoutis X, Del Fiore JM, Thubert P, Montavont N. From best effort to deterministic packet delivery for wireless industrial IoT networks. *IEEE Transactions on Industrial Informatics* 2018; 14(10): 4468–4480.
10. Dujovne D, Watteyne T, Vilajosana X, Thubert P. 6TiSCH: deterministic IP-enabled industrial internet (of things). *IEEE Communications Magazine* 2014; 52(12): 36-41. doi: 10.1109/MCOM.2014.6979984
11. Duquennoy S, Elsts A, Al Nahas B, Oikonomou G. TSCH and 6TiSCH for Contiki: Challenges, Design and Evaluation. In: 2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS). ; 2017
12. Spörk M, Boano CA, Zimmerling M, Römer K. BLEach: Exploiting the Full Potential of IPv6 over BLE in Constrained Embedded IoT Devices. In: Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems. ; 2017
13. Charlier M, Quoitin B, Hauweele D. UWB-TSCH : Time and Frequency Division Multiplexing for UWB Communications. In: CoRes 2019. ; 2019; Saint Laurent de la Cabrerisse, France.
14. Bor M, Vidler J, Roedig U. LoRa for the Internet of Things. In: Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks. Junction Publishing; 2016; USA: 361–366.
15. Piyare R, Murphy AL, Magno M, Benini L. KRATOS: An Open Source Hardware-Software Platform for Rapid Research in LPWANs. In: 14th Int Conf Wireless and Mobile Computing, Networking and Communications (WiMob). ; 2018: 1-4
16. Dias J, Grilo A. LoRaWAN Multi-hop Uplink Extension. *Procedia Comput. Sci.* 2018; 130(C): 424–431.
17. Duquennoy S, Al Nahas B, Landsiedel O, Watteyne T. Orchestra: Robust Mesh Networks Through Autonomously Scheduled TSCH. In: Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems. ; 2015: 337–350
18. Elsts A, Duquennoy S, Fafoutis X, Oikonomou G, Piechocki R, Craddock I. Microsecond-Accuracy Time Synchronization Using the IEEE 802.15.4 TSCH Protocol. In: 2016 IEEE 41st Conference on Local Computer Networks Workshops (LCN Workshops). ; 2016: 156-164
19. Rizzi M, Ferrari P, Flammini A, Sisinni E, Gidlund M. Using LoRa for industrial wireless networks. In: 2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS). ; 2017: 1-4
20. Haxhibeqiri J, Abeele V. dF, Moerman I, Hoebeke J. LoRa scalability: A simulation model based on interference measurements. *Sensors* 2017; 17(6): 1193.